AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [0106] with the following amended paragraph:

In addition to implementations of the invention using hardware, the invention can be implemented in computer readable code (e.g., computer readable program code, data, etc.) embodied in a computer usable (e.g., readable) medium. The computer code causes the enablement of the functions or fabrication or both of the invention disclosed herein. For example, this can be accomplished through the use of general programming languages (e.g., C, C++, JAVA, and the like); GDSII databases; hardware description languages (HDL) including Verilog HDL, VHDL, Altera HDL (AHDL), and so on; or other programming and/or circuit (i.e., schematic) capture tools available in the art. The computer code can be disposed in any known computer usable (e.g., readable) medium including semiconductor memory, magnetic disk, optical disk (e.g., CD-ROM, DVD-ROM, and the like), and as a computer data signal embodied in a computer usable (e.g., readable) transmission medium (e.g., ~~carrier wave or any other~~ medium including digital, optical or analog-based medium). As such, the computer code can be transmitted over communication networks, including Internets and intranets. It is understood that the invention can be embodied in computer code (e.g., as part of an IP (intellectual property) core, such as a microprocessor core, or as a system-level design, such as a System on Chip (SOC)) and transformed to hardware as part of the production of integrated circuits. Also, the invention may be embodied as a combination of hardware and computer code.

Please delete the section entitled "SUMMARY OF THE INVENTION" in its entirety and substitute the following section therefor:

SUMMARY OF THE INVENTION

The present invention provides an apparatus for killing an instruction loaded into an instruction queue of a microprocessor during a first clock cycle and output from a bottom entry of the instruction queue during a second clock cycle subsequent to the first clock cycle. The apparatus includes a kill signal, for conveying a value generated during a third clock cycle subsequent to the first clock cycle. The apparatus also includes a kill queue, coupled to the kill signal, for loading the kill signal value generated during the

third clock cycle, and for outputting the kill signal value during the second clock cycle. The apparatus also includes a load signal, coupled to the kill queue, which indicates during the second clock cycle whether the instruction was loaded into the bottom entry of the instruction queue during the first clock cycle. If the load signal is true, the third clock cycle and the second clock cycle are a same clock cycle. The apparatus also includes a valid signal, coupled to the kill queue, generated during the second clock cycle for indicating whether the instruction is to be executed by the microprocessor. The valid signal is false if the kill signal value output by the kill queue during the second clock cycle is true.

In another aspect, the present invention provides a method for killing an instruction in a microprocessor. The method includes loading an instruction into a first queue during a first clock cycle, generating a kill signal during a second clock cycle subsequent to the first clock cycle, and loading a value of the kill signal into a second queue during the second clock cycle. The method also includes determining whether the value in the second queue is true during a third clock cycle in which the instruction is output from a bottom entry of the first queue, and foregoing execution of the instruction if the value is true. The method also includes generating a load signal for indicating during the third clock cycle whether the instruction was loaded into a bottom entry of the first queue during the first clock cycle. If the load signal is true, the third clock cycle and the second clock cycle are a same clock cycle.

In another aspect, the present invention provides a program embodied on a computer readable medium, including computer-readable program code for providing an apparatus for killing an instruction loaded into an instruction queue of a microprocessor during a first clock cycle and output from a bottom entry of the instruction queue during a second clock cycle subsequent to the first clock cycle. The program code includes first program code for providing a kill signal, for conveying a value generated during a third clock cycle subsequent to the first clock cycle. The program code also includes second program code for providing a kill queue, coupled to the kill signal, for loading the kill signal value generated during the third clock cycle, and for outputting the kill signal value during the second clock cycle. The program code also includes third program code

for providing a load signal, coupled to the kill queue, for indicating during the second clock cycle whether the instruction was loaded into the bottom entry of the instruction queue during the first clock cycle. If the load signal is true, the third clock cycle and the second clock cycle are a same clock cycle. The program code also includes fourth program code for providing a valid signal, coupled to the kill queue, generated during the second clock cycle for indicating whether the instruction is to be executed by the microprocessor. The valid signal is false if the kill signal value output by the kill queue during the second clock cycle is true.

In another aspect, the present invention provides an apparatus for killing an instruction loaded into an instruction queue of a microprocessor during a first clock cycle and output from a bottom entry of the instruction queue during a second clock cycle subsequent to the first clock cycle. The apparatus includes a kill signal, for conveying a value generated during a third clock cycle subsequent to the first clock cycle. The apparatus also includes a kill queue, coupled to the kill signal, for loading the kill signal value generated during the third clock cycle, and for outputting the kill signal value during the second clock cycle. The apparatus also includes a load signal, coupled to the kill queue, which indicates during the second clock cycle whether the instruction was loaded into the bottom entry of the instruction queue during the first clock cycle. If the load signal is false, the second clock cycle is subsequent to the third clock cycle. The apparatus also includes a valid signal, coupled to the kill queue, generated during the second clock cycle for indicating whether the instruction is to be executed by the microprocessor. The valid signal is false if the kill signal value output by the kill queue during the second clock cycle is true.

In another aspect, the present invention provides a method for killing an instruction in a microprocessor. The method includes loading an instruction into a first queue during a first clock cycle, generating a kill signal during a second clock cycle subsequent to the first clock cycle, and loading a value of the kill signal into a second queue during the second clock cycle. The method also includes determining whether the value in the second queue is true during a third clock cycle in which the instruction is output from a bottom entry of the first queue, and foregoing execution of the instruction if the value is

true. The method also includes generating a load signal for indicating during the third clock cycle whether the instruction was loaded into a bottom entry of the first queue during the first clock cycle. If the load signal is false, the third clock cycle is subsequent to the second clock cycle.

In another aspect, the present invention provides a program embodied on a computer readable medium, including computer-readable program code for providing an apparatus for killing an instruction loaded into an instruction queue of a microprocessor during a first clock cycle and output from a bottom entry of the instruction queue during a second clock cycle subsequent to the first clock cycle. The program code includes first program code for providing a kill signal, for conveying a value generated during a third clock cycle subsequent to the first clock cycle. The program code also includes second program code for providing a kill queue, coupled to the kill signal, for loading the kill signal value generated during the third clock cycle, and for outputting the kill signal value during the second clock cycle. The program code also includes third program code for providing a load signal, coupled to the kill queue, for indicating during the second clock cycle whether the instruction was loaded into the bottom entry of the instruction queue during the first clock cycle. If the load signal is false, the second clock cycle is subsequent to the third clock cycle. The program code also includes fourth program code for providing a valid signal, coupled to the kill queue, generated during the second clock cycle for indicating whether the instruction is to be executed by the microprocessor. The valid signal is false if the kill signal value output by the kill queue during the second clock cycle is true.

An advantage of the present invention is that it enables proper program execution in a microprocessor pipeline that employs instruction queues and means requiring instruction killing, such as branch prediction mechanisms. Another advantage is that the present invention allows the kill signal to be generated late without adding additional pipeline stages to accommodate the instruction queue.

Other features and advantages of the present invention will become apparent upon study of the remaining portions of the specification and drawings.